université de BORDEAUX

Formation CNRS
18 Novembre 2016
Python pour la biologie

Biopython: Accessing NCBI's Entrez databases

cgfb
BIOINFORMATIQUE

université de BORDEAUX

Entrez Guidelines

- Be sensible with your usage levels. If you plan to download lots of data, consider other options.
- For example, if you want easy access to all the human genes, consider fetching each chromosome by FTP as a GenBank file, and importing these into your own BioSQL database.

- Don’t forget to import the module Entrez

```
>>> from Bio import Entrez
```

EInfo: Obtaining information about the Entrez databases

➔ EInfo provides field index term counts, last update, and available links for each of NCBI's databases

```
>>> Entrez.email = "A.N.Other@example.com" # Always tell NCBI who you are
>>> handle = Entrez.einfo()
>>> result = handle.read()
>>> print(result) <?xml version="1.0"?>
<!DOCTYPE eInfoResult PUBLIC "-//NLM//DTD eInfoResult, 11 May 2002//EN"
"http://www.ncbi.nlm.nih.gov/entrez/query/DTD/eInfo_020511.dtd">
<eInfoResult>
<DbList>
<DbName>pubmed</DbName>
<DbName>protein</DbName>
<DbName>nucleotide</DbName>
<DbName>nuccore</DbName>
<DbName>nucgss</DbName>
……
<DbName>genome</DbName>
<DbName>books</DbName>
<DbName>cancerchromosomes</DbName>
<DbName>unigene</DbName>
<DbName>unists</DbName>
</DbList>
</eInfoResult>
```

EInfo: Obtaining information about the Entrez databases

➔ Using Bio.Entrez's parser instead, we can directly parse this XML file into a Python object:

```
>>> handle = Entrez.einfo()
>>> record = Entrez.read(handle)
```

➔ Now record is a dictionary with exactly one key

```
>>> record.keys()
[u'DbList']
```

➔ The values stored in this key is the list of database names shown in the XML above:

```
>>> record["DbList"]
['pubmed', 'protein', 'nucleotide', 'nuccore', 'nucgss', 'nucest',
'structure', 'genome', 'books', 'cancerchromosomes', 'cdd', 'gap',
'domains', 'gene', 'genomeprj', 'gensat', 'geo', 'gds', 'homologene',
'journals', 'mesh', 'ncbisearch', 'nlmcatalog', 'omia', 'omim', 'pmc',
'popset', 'probe', 'proteinclusters', 'pcassay', 'pccompound',
'pcsubstance', 'snp', 'taxonomy', 'toolkit', 'unigene', 'unists']
```

EInfo: Obtaining information about the Entrez databases

➔ For each of these databases, we can use EInfo again to obtain more information:

```
>>> handle = Entrez.einfo(db="pubmed")
>>> record = Entrez.read(handle)
>>> record["DbInfo"]["Description"]
'PubMed bibliographic record'
>>> record["DbInfo"]["Count"]
'17989604'
>>> record["DbInfo"]["LastUpdate"]
'2008/05/24 06:45'
```

➔ Try record["DbInfo"].keys() for other information stored in this record. One of the most useful is a list of possible search fields for use with ESearch

```
>>> for field in record["DbInfo"]["FieldList"]:
... print("%(Name)s, %(FullName)s, %(Description)s" % field)
ALL, All Fields, All terms from all searchable fields
UID, UID, Unique number assigned to publication
FILT, Filter, Limits the records
TITL, Title, Words in title of publication
WORD, Text Word, Free text associated with publication
AUTH, Author, Author(s) of publication
JOUR, Journal, Journal abbreviation of publication
...
```

ESearch: Searching the Entrez databases

- To search any of these databases, we use Bio.Entrez.esearch()
- let's search in PubMed for publications related to Biopython

```
>>> from Bio import Entrez
>>> Entrez.email = "A.N.Other@example.com" # Always tell NCBI who you are
>>> handle = Entrez.esearch(db="pubmed", term="biopython")
>>> record = Entrez.read(handle)
>>> record["IdList"]
['19304878', '18606172', '16403221', '16377612', '14871861', '14630660', '12230038']
```

- 7 PubMed IDs (including 19304878 which is the PMID for the Biopython application note), which can be retrieved by EFetch

```
>>> handle = Entrez.esearch(db="nucleotide", term="Cypripedioideae[Orgn] AND matK[Gene]")
>>> record = Entrez.read(handle)
>>> record["Count"]
'25'
>>> record["IdList"]
['126789333', '37222967', '37222966', '37222965', ..., '61585492']
```

ESearch: Searching the Entrez databases

- Note that instead of a species name like Cypripedioideae[Orgn], you can restrict the search using an NCBI taxon identifier, here this would be txid158330[Orgn]
- For example, including complete[prop] in a genome search restricts to just completed genomes
- let's get a list of computational journal titles

```
>>> handle = Entrez.esearch(db="nlmcatalog", term="computational[Journal]", retmax='20')
>>> record = Entrez.read(handle)
>>> print("{} computational journals found".format(record["Count"]))
117 computational Journals found
>>> print("The first 20 are\n{}".format(record['IdList']))
['101660833', '101664671', '101661657', '101659814', '101657941',
'101653734', '101669877', '101649614', '101647835', '101639023',
'101627224', '101647801', '101589678', '101585369', '101645372',
'101586429', '101582229', '101574747', '101564639', '101671907']
```

Epost: Uploading a list of identifiers

- EPost uploads a list of UIs for use in subsequent search strategies
- It is available from Biopython through the Bio.Entrez.epost() function

```
>>> from Bio import Entrez
>>> Entrez.email = "A.N.Other@example.com" # Always tell NCBI who you are
>>> id_list = ["19304878", "18606172", "16403221", "16377612", "14871861", "14630660"]
>>> print(Entrez.epost("pubmed", id=",".join(id_list)).read())
<?xml version="1.0"?>
<!DOCTYPE ePostResult PUBLIC "-//NLM//DTD ePostResult, 11 May 2002//EN"
"http://www.ncbi.nlm.nih.gov/entrez/query/DTD/ePost_020511.dtd">
<ePostResult>
<QueryKey>1</QueryKey>
<WebEnv>NCID_01_206841095_130.14.22.101_9001_1242061629</WebEnv>
</ePostResult>
```

- XML includes two important strings:
  - QueryKey and WebEnv which together de ne your history session.
  - You would extract these values for use with another Entrez call such as EFetch:

```
>>> from Bio import Entrez
>>> Entrez.email = "A.N.Other@example.com" # Always tell NCBI who you are
>>> id_list = ["19304878", "18606172", "16403221", "16377612", "14871861", "14630660"]
>>> search_results = Entrez.read(Entrez.epost("pubmed", id=",".join(id_list)))
>>> webenv = search_results["WebEnv"]
>>> query_key = search_results["QueryKey"]
```

ESummary: Retrieving summaries from primary IDs

- ESummary retrieves document summaries from a list of primary IDs
- In Biopython, ESummary is available as Bio.Entrez.esummary()

```
>>> from Bio import Entrez
>>> Entrez.email = "A.N.Other@example.com" # Always tell NCBI who you are
>>> handle = Entrez.esummary(db="nlmcatalog", id="101660833")
>>> record = Entrez.read(handle)
>>> info = record[0]['TitleMainList'][0]
>>> print("Journal info\nid: {}\nTitle: {}".format(record[0]["Id"], info["Title"]))
Journal info
id: 101660833
Title: IEEE transactions on computational imaging.
```

EFetch: Downloading full records from Entrez

- Use when you want to retrieve a full record from Entrez.
- For most of their databases, the NCBI support several different formats
- Requires specifying the rettype and/or retmode optional arguments

```
>>> from Bio import Entrez
>>> Entrez.email = "A.N.Other@example.com" # Always tell NCBI who you are
>>> handle = Entrez.efetch(db="nucleotide", id="186972394", rettype="gb", retmode="text")
>>> print(handle.read())
LOCUS EU490707 1302 bp DNA linear PLN 05-MAY-2008
DEFINITION Selenipedium aequinoctiale maturase K (matK) gene, partial cds;
chloroplast.
ACCESSION EU490707
VERSION EU490707.1 GI:186972394
KEYWORDS .
SOURCE chloroplast Selenipedium aequinoctiale
ORGANISM Selenipedium aequinoctiale
Eukaryota; Viridiplantae; Streptophyta; Embryophyta; Tracheophyta;
Spermatophyta; Magnoliophyta; Liliopsida; Asparagales; Orchidaceae;
Cypripedioideae; Selenipedium.
REFERENCE 1 (bases 1 to 1302)
AUTHORS Neubig,K.M., Whitten,W.M., Carlsward,B.S., Blanco,M.A.,
Endara,C.L., Williams,N.H. and Moore,M.J.
TITLE Phylogenetic utility of ycf1 in orchids
JOURNAL Unpublished
REFERENCE 2 (bases 1 to 1302)
AUTHORS Neubig,K.M., Whitten,W.M., Carlsward,B.S., Blanco,M.A.,
```

GenBank record 186972394

TITLE Direct Submission
JOURNAL Submitted (14-FEB-2008) Department of Botany, University of Florida, 220 Bartram Hall, Gainesville, FL 32611-8526, USA
FEATURES Location/Qualifiers
source 1..1302
/organism="Selenipedium aequinoctiale"
/organelle="plastid:chloroplast"
/mol_type="genomic DNA"
/specimen_voucher="FLAS:Blanco 2475"
/db_xref="taxon:256374"
gene <1..>1302
/gene="matK"
CDS <1..>1302
/gene="matK"
/codon_start=1
/transl_table=11
/product="maturase K"
/protein_id="ACC99456.1"
/db_xref="GI:186972395"
/translation="IFYEPVEIFGYDNKSSLVLVKRLITRMYQQNFLISSVNDSNQKG
FWGHKHFFSSHFSSQMVSEGFGVILEIPFSSQLVSSLEEKKIPKYQNLRSIHSIFPFL
EDKFLHLNYVSDLLIPHPIHLEILVQILQCRIKDVPSLHLLRLLFHEYHNLNSLITSK
KFIYAFSKRKKRFLWLLYNSYVYECEYLFQFLRKQSSYLRSTSSGVFLERTHLYVKIE

GenBank record 186972394

HLLVVCCNSFQRILCFLKDPFMHYVRYQGKAILASKGTLILMKKWKFHLVNFWQSYFH
FWSQPYRIHIKQLSNYSFSFLGYFSSVLENHLVVRNQMLENSFIINLLTKKFDTIAPV
ISLIGSLSKAQFCTVLGHPISKPIWTDFSDSDILDRFCRICRNLCRYHSGSSKKQVLY
RIKYILRLSCARTLARKHKSTVRTFMRRLGSGLLEEFFMEEE"
ORIGIN
1 atttttacg aacctgtgga aatttttggt tatgacaata aatctagttt agtacttgtg
61 aaacgtttaa ttactcgaat gtatcaacag aatttttga tttcttcggt taatgattct
121 aaccaaaaag gattttgggg gcacaagcat ttttttctt ctcatttttc ttctcaaatg
181 gtatcagaag gttttggagt cattctggaa attccattct cgtcgcaatt agtatcttct
241 cttgaagaaa aaaaaatacc aaaatatcag aatttacgat ctattcattc aatatttccc
301 tttttagaag acaaattttt acatttgaat tatgtgtcag atctactaat accccatccc
361 atccatctgg aaatcttggt tcaaatcctt caatgccgga tcaaggatgt tccttctttg
421 catttattgc gattgctttt ccacgaatat cataatttga atagtctcat tacttcaaag
481 aaattcattt acgccttttc aaaaagaaag aaaagattcc tttggttact atataattct
541 tatgtatatg aatgcgaata tctattccag tttcttcgta aacagtcttc ttatttacga
601 tcaacatctt ctggagtctt tcttgagcga acacatttat atgtaaaaat agaacatctt
661 ctagtagtgt gttgtaattc ttttcagagg atcctatgct ttctcaagga tcctttcatg
721 cattatgttc gatatcaagg aaaagcaatt ctggcttcaa agggaactct tattctgatg
781 aagaaatgga aatttcatct tgtgaatttt tggcaatctt attttcactt ttggtctcaa
841 ccgtatagga ttcatataaa gcaattatcc aactattcct tctcttttct ggggtatttt
901 tcaagtgtac tagaaaatca tttggtagta agaaatcaaa tgctagagaa ttcatttata
961 ataaatcttc tgactaagaa attcgatacc atagccccag ttatttctct tattggatca
1021 ttgtcgaaag ctcaattttg tactgtattg ggtcatccta ttagtaaacc gatctggacc
1081 gatttctcgg attctgatat tcttgatcga ttttgccgga tatgtagaaa tctttgtcgt
1141 tatcacagcg gatcctcaaa aaaacaggtt ttgtatcgta taaaatatat acttcgactt
1201 tcgtgtgcta gaactttggc acggaaacat aaaagtacag tacgcacttt tatgcgaaga
1261 ttaggttcgg gattattaga agaattcttt atggaagaag aa

Parse it into a seq record

```
>>> from Bio import Entrez, SeqIO
>>> handle = Entrez.efetch(db="nucleotide", id="186972394", rettype="gb", retmode="text")
>>> record = SeqIO.read(handle, "genbank")
>>> handle.close()
>>> print(record)
ID: EU490707.1
Name: EU490707
Description: Selenipedium aequinoctiale maturase K (matK) gene, partial cds; chloroplast.
Number of features: 3
...
Seq('ATTTTTTACGAACCTGTGGAAATTTTTGGTTATGACAATAAATCTAGTTTAGTA...GAA', IUPACAmbiguousDNA())
```

Save the sequence data to a local file

- Typical use would be to save the sequence data to a local file and then parse it with Bio.SeqIO
- Save you to re-download the same file repeatedly

```
import os
from Bio import SeqIO
Entrez.email = "A.N.Other@example.com" # Always tell NCBI who you are
filename = "gi_186972394.gbk"
if not os.path.isfile(filename):
print(“Downloading...”)
net_handle = Entrez.efetch(db="nucleotide", id="186972394", rettype="gb", retmode="text")
out_handle = open(filename, "w")
out_handle.write(net_handle.read())
out_handle.close()
net_handle.close()
print("Saved")
print("Parsing...")
record = SeqIO.read(filename, "genbank")
print(record)
```

```
>>> handle = Entrez.efetch(db="nucleotide", id="186972394", retmode="xml")
>>> record = Entrez.read(handle)
>>> handle.close()
>>> record[0]["GBSeq_definition"]
'Selenipedium aequinoctiale maturase K (matK) gene, partial cds; chloroplast'
>>> record[0]["GBSeq_source"]
'chloroplast Selenipedium aequinoctiale’
```

ELink: Searching for related items in NCBI Entrez

➔ Find related items in the NCBI Entrez database

```
>>> from Bio import Entrez
>>> Entrez.email = "A.N.Other@example.com"
>>> pmid = "19304878"
>>> record = Entrez.read(Entrez.elink(dbfrom="pubmed", id=pmid))
```

➔ a Python list, one for each database in which we searched

```
>>> record[0]["DbFrom"]
'pubmed'
>>> record[0]["IdList"]
['19304878']
>>> len(record[0]["LinkSetDb"])
5
>>> for linksetdb in record[0]["LinkSetDb"]:
... print(linksetdb["DbTo"], linksetdb["LinkName"], len(linksetdb["Link"]))
pubmed pubmed_pubmed 110
pubmed pubmed_pubmed_combined 6
pubmed pubmed_pubmed_five 6
pubmed pubmed_pubmed_reviews 5
pubmed pubmed_pubmed_reviews_five 5
```

➔ The actual search results are stored as under the "Link” key

```
>>> record[0]["LinkSetDb"][0]["Link"][0]
{u'Id': '19304878'}
```

➔ let's look at the second search result

```
>>> record[0]["LinkSetDb"][0]["Link"][1]
{u'Id': '14630660'}
```

➔ This paper, with PubMed ID 14630660, is about the Biopython PDB parser.

```
>>> for link in record[0]["LinkSetDb"][0]["Link"]:
... print(link["Id"])
19304878
14630660
18689808
17121776
16377612
12368254
```

EGQuery: Global Query - counts for search terms

```
>>> from Bio import Entrez
>>> Entrez.email = "A.N.Other@example.com" # Always tell NCBI who you are
>>> handle = Entrez.egquery(term="biopython")
>>> record = Entrez.read(handle)
>>> for row in record["eGQueryResult"]:
... print(row["DbName"], row["Count"])
...
pubmed 6
pmc 62
journals 0
...
```

ESpell: Obtaining spelling suggestions

```
>>> from Bio import Entrez
>>> Entrez.email = "A.N.Other@example.com" # Always tell NCBI who you are
>>> handle = Entrez.espell(term="biopythooon")
>>> record = Entrez.read(handle)
>>> record["Query"]
'biopythooon'
>>> record["CorrectedQuery"]
'biopython'
```